

年终总结

项目经历

一、影像数据生产任务监控、分发、管理系统-DMES(Data MES,数据生产MES系统)的重构和更新.(重构经验)

贡献:

1. 代码架构调整。参与前端的基础设施的技术重选型,参与前端代码架构设计与实现 按照新版本代码架构重新封装API层代码,规范组件名称,将组件式路由调整为递归函数生成的配置式路由。
2. 业务逻辑重新实现同时按照需求更新。

项目难点:

1. 业务逻辑较为繁重,大致分为: 物料(数据约束) 工序(数据处理单元) 工艺(数据生产约束和流程) 调试工艺(测试) 项目(生产) 入库(数据管理) 在和后端对接接口的过程中, 参考后端业务模型, 前端建立对应的类(数据层抽象封装),与后端抽象保持一致,提升代码可维护性。
2. 权限系统分权分域设计:

前端建立Purview权限的数据模型(类),实现 1.原子权限值通过按位与(&)操作组合成业务逻辑权限(某特定业务操作的权限) 2.用户权限总值结合用户所在域与业务逻辑权限进行按位与(&)操作判断用户是否具有该业务权限

3. 根据后端的JSON生成表单,动态的表单项,动态的校验规则:

二次封装Form.Item组件,根据JSON递归生成Form.Item,

4. 维护过程中,阅读AntV X6源码 优化由AntV X6编写的工艺流程图添加节点时的性能,理解其性能差距的来源,canvas模式与svg模式的不同,后续将X6升级为G6.
5. 根据客户要求进行定制开发时需求的变动导致代码的变动,业务组件的可维护性至关重要,不停反思组件的封装思路,增强组件的可维护性,思考到底什么是真正"好的封装"

二、遥感影像轻型检索系统(业务经验)

遥感影像检索、浏览、管理系统

贡献

前端部分由单人独立完成

1. 技术选型:脚手架:customize-cra + react-app-rewired + cra 地图框架:leaflet UI框架: antd react 状态管理工具: dva
2. 独立完成,封装'数据驱动'风格的leaflet图层组件,灵活运用react 组件的生命周期,通过组件组合的方式实现图层增删与更新的操作,完成(地图框架副作用)和react状态的同步

项目难点:

1. 遥感影像数据模型(业务模型),中类型tif数据或者其他传统遥感数据中的元数据(metaData)的理解和数据模型的封装,定义MetalInfo 类,并封装提取Bounds的方法(用于图层外接矩形高亮或者数据定位),还要配合后端接口的设计,封装一系列的'工具'方法.
2. 地图库Leaflet和react库配合使用时, 不断强化了对React设计中的"副作用"的理解,以及RFC hooks的理解有一定的要求,自此开始使用RFC作为自己开发组件的主要方式。
3. 解析wmmts,wms地图服务中的xml中的参数,在实例化图层的时候作为图层参数为图层增加属性(如中心点,加载范围,外接矩形,优化前端展示图层时的性能)。

三、地理大数据仓库管理平台,负责数据浏览模块。(微前端团队管理经验)

贡献:

项目使用了umi + dva(umi自带的状态管理)精简项目配置过程。团队采用微前端(qiankun)的开发模式,多个模块技术栈分离并独立开发,最终逐步集成。

- **业务方向:** 帮助县级、市级、省级卫星应用中心的用户解决了矢量、遥感及描述文件或相关文档等空间大数据的管理、运转、出库入库效率低下,管理粗糙的问题。
- **研发模式:** 使用微前端架构,提升了开发效率和集成效率,研发速度加快。
- **交互亮点:** 数据浏览页面是文件目录+目录列表构成的 类似windows资源管理器的左右双栏布局,用户'当前所在目录'这一状态决定整个视图的数据,这是实现目录和目录下文件列表的ui的数据驱动视图核心。

自此个人心智重心开始从技术实现细节 转向 业务逻辑和系统设计理解,辅以具体技术用于调试和学习源码中的设计模式,丰富自己的经验,不断提升遇到框架issue中的问题的解决能力。

四、卫星数据应用中心, 数字化大屏项目(可视化经验)

贡献: 在团队成员对echarts不熟悉并且ui休产假的情况下, 承担实现技巧突破手的角色, 先分工, 最后整合的时候修复同事代码的bug, 在规定时间内推动项目完成, 项目验收期间赴北京生态卫星中心, 现场根据用户需求调整设计, 一个人维护三个同事的代码, 客户对开发效率很满意。

TODO

"understand why it is the way it is." —— 《Speaking JavaScript: An In-Depth Guide for Programmers》Dr. Axel Rauschmayer

- 跳出GIS领域, 做一个有gis开发能力的开发者, 而不仅仅做一个限定在gis领域的开发者 迫切需要一个平台, 去学习去挖掘, 成就他人的同时成就自己。
- 探索"为什么工具被设计成这样,它解决了什么问题,他的核心机制是什么"反而更有意思,世界是变化的,探索变化过程中的不变的东西才是最重要的,也是从长期来看对团队收益最大的(当然在不影响团队合作进度的前提下去自由探索)
- 重学JS、刷题,1~2个月1本,详情见书单
- 最近看到同济大学的 Intelligent Big Data Visualization Lab 对他们在数据可视化方向的研究很感兴趣